

## Research Article

# Service Outsourcing Character Oriented Privacy Conflict Detection Method in Cloud Computing

Changbo Ke,<sup>1,2</sup> Zhiqiu Huang,<sup>1</sup> Weiwei Li,<sup>1</sup> Yi Sun,<sup>1</sup> and Fangxiong Xiao<sup>1,3</sup>

<sup>1</sup> College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, Jiangsu 210016, China

<sup>2</sup> School of Computer Science & Technology/School of Software, Nanjing University of Posts and Telecommunications, Nanjing, Jiangsu 210023, China

<sup>3</sup> School of Information and Statistics, Guangxi University of Finance and Economics, Nanning 530003, China

Correspondence should be addressed to Changbo Ke; [brobo.ke@gmail.com](mailto:brobo.ke@gmail.com)

Received 21 November 2013; Accepted 23 February 2014; Published 15 May 2014

Academic Editor: Zhi-Hong Guan

Copyright © 2014 Changbo Ke et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Cloud computing has provided services for users as a software paradigm. However, it is difficult to ensure privacy information security because of its opening, virtualization, and service outsourcing features. Therefore how to protect user privacy information has become a research focus. In this paper, firstly, we model service privacy policy and user privacy preference with description logic. Secondly, we use the pellet reasoner to verify the consistency and satisfiability, so as to detect the privacy conflict between services and user. Thirdly, we present the algorithm of detecting privacy conflict in the process of cloud service composition and prove the correctness and feasibility of this method by case study and experiment analysis. Our method can reduce the risk of user sensitive privacy information being illegally used and propagated by outsourcing services. In the meantime, the method avoids the exception in the process of service composition by the privacy conflict, and improves the trust degree of cloud service providers.

## 1. Introduction

Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction [1]. With the character of service outsourcing, virtualization, distribution, and multitenancy, cloud computing has become a new computing paradigm and research focus. Such characters enhance the service quality and reduce the wastage of computing resources; for example, service outsourcing enhances the service capability and specialization through service composition [2]. Because of the transparency of privacy information to the outsourcing service provider, users worry that it will be hard to prevent user privacy data from being illegally propagated and used. For example, Google is sued by many users in America because of its new unified privacy policy implemented from March 1st, 2012. In Europe, the implementation of this new privacy policy

has been investigated by European Union and postponed. According to the analysis by America Electronic Privacy Information Center, Google's new privacy policies do not consider how to use privacy data in the product and to whom privacy data is propagated according to user privacy requirement and these policies may have conflicts with local laws. Therefore, privacy protection in cloud computing has become research focus in evolving computing paradigm.

Privacy was proposed as the human right to be let alone in the beginning [3]. In the domain of information system and software engineering, privacy protection means the capability of preventing individual information from being collected, disclosed, and stored by others [4]. The Platform for Privacy Preferences (P3P) [5] developed by World Wide Consortium (W3C) in 2002 provides a standard and machine-understandable privacy policy, which matches with user privacy preference. According to the matched results, user can select service that meets privacy preference. However, the described privacy requirement in P3P lacks semantic information and P3P only applies to web site,

not supporting privacy protection in service composition. Therefore P3P cannot be applied in cloud computing since all entities in cloud computing are service and provide service through service composition. In 2005 Extensible Access Control Markup Language (XACML) [6] is proposed by organization of the Advancement of Structured Information Standards (OASIS). XACML 2.0 [7] extends the support of privacy policy through profile for privacy policies. However, different users in cloud computing have different privacy requirements requiring different definition of sensitive privacy information. XACML privacy policies only apply to service provider without considering user privacy requirement and hardly guarantee the composite service satisfying user privacy requirement. Pearson [8] defined privacy protection in cloud computing as the capability of user to control personal sensitive information (PSI) without being collected, used, disclosed, and stored by cloud service provider. Pearson et al. [9, 10] proposed a conception of accountability that can create solutions to support users in deciding and tracking how their data is used by cloud service providers. They provided certain theoretical guidance but did not put forward specific solution about privacy protection in cloud computing. Roy et al. [11] and Bowers et al. [12] executed different privacy protection policy for data at different implementing stage in cloud computing. Moreover, privacy protection policy is integrated in services and privacy protection executor is service provider. Therefore, service provider is hardly arbitrated when user privacy information is illegally disclosed.

In cloud computing all entities are services. To satisfy and execute user privacy requirement properly, user privacy protection service must be provided by third party. Therefore, we propose a method of building service of privacy conflict detection in cloud computing. Suppose service document in cloud computing is described with OWL-S. In this paper, we firstly obtain input and precondition of service from service description document, model the input and precondition of service by taking advantage of TBox in description logic, and model user privacy preference by using ABox in description logic. In this way, we get the knowledge base. Then we reason the knowledge base with description logic, namely, taking advantage of Tableau algorithm to verify the consistency and satisfiability of the knowledge base, so as to detect the conflict between input and precondition of services and user privacy preferences policy. In this way we can ensure user privacy right and supervise privacy information propagating among outsourcing services. At last, we present the algorithm of detecting privacy conflict that supports semantics in cloud computing and prove the correctness and feasibility of this method by case study and experiment analysis.

## 2. Related Works

We classify the related works of privacy protection into two parts which are computing process oriented privacy protection and data oriented privacy protection. The first part is classified into five smaller parts, which are model and verification of privacy requirement, matching and negotiation

of privacy policy, and disclosure and risk. The second part is classified into three smaller parts, which are obfuscation, encryption, and anonymity of privacy data. In the meantime, we organize the related works into tables and compare them from contributions, applied computing paradigm, whether supporting service composition and whether supporting semantics. We highlight our work in the tables in detailed contents as shown in Table 1.

Since our work is focusing on privacy policy matching, we majorly discuss the related works of this theme. Other related works are organized into tables without further discussion. Barth et al. [17] defined user and service provider privacy policy, respectively, on the basis of analyzing current privacy rules and proposed a privacy policy automatic matching method, which can check the type of privacy data, the objective of privacy data disclosure, the collector, and maintenance period of privacy data. Wei et al. [18] researched privacy data protection policy in application of pervasive computing, built privacy model, and privacy policy axiom by using many-sorted logic and description logic and proposed a reason method of privacy policy which can check the inconsistency among policies.

## 3. Motivation

To explicitly clarify our research issue, we present an application scenario as follows.

Suppose Tom wants to buy commodity from seller *B* through service *A* in cloud computing; service *A* requires Tom to input his sensitive privacy information, like real name, bank account, mobile phone number, and detailed address. Without negotiation with service *A* for privacy agreement, Tom may worry about two aspects.

- (1) Privacy information may be illegally used or propagated by service *A* or seller *B*. Because of no privacy agreement, Tom cannot sue service *A* or seller *B* for recovering financial or spiritual losses. If Tom does not eagerly want this service, once he has privacy conflict with service provider, Tom will stop the service and select other services. Scenario is shown in Figure 1(a).
- (2) If Tom eagerly wanted to obtain the service, he would provide sensitive privacy information to service *A*. However, privacy information is disclosed causing financial or spiritual losses. Scenario is shown in Figure 1(b).

In this paper, our motivation is to build a service, which automatically provides conflict detection of privacy for both user and service provider in cloud computing. Through this service, services satisfying user privacy requirement are discovered, so as to protect user privacy information without being illegally used and propagated.

## 4. Basic Theories

*4.1. Description Logic Basis.* Description logic is the basis of Ontology Web Language for Service (OWL-S), which

TABLE 1: Comparison of related works.

Methods	Authors	Contributions	Computing paradigm	Support for service composition	Support for semantic
Modeling and verification	Hamadi et al. [13]	Conceptual modeling of privacy-aware web service	Service computing	✗	✓
	Guermouche et al. [14]	Privacy-aware web service protocol replaceability	Service computing	✗	✓
	Mokhtari et al. [15]	Verification of privacy timed properties	Service computing	✗	✗
	Lin Yuan et al. [16]	Minimal privacy authorization in web services collaboration	Service computing	✓	✗
Matching	Barth et al. [17]	Privacy and utility in business processes	Service computing	✓	✓
	Wei et al. [18]	Privacy-protection policy for pervasive computing	Pervasive computing	✗	✓
	<b>Our work</b>	<b>Outsourcing service oriented privacy conflict detection Method in cloud computing</b>	<b>Cloud Computing</b>	✓	✓
Negotiation	Zhu and Zhou [19]	Role-based collaboration and its kernel mechanisms	N/A	N/A	✗
	El-Khatib [20]	A privacy negotiation protocol for web services	Service Computing	✗	✗
	Zhang and Fen [21]	Parsimonious semantic trust negotiation	N/A	N/A	✓
Disclosure	Kolter et al. [22]	Visualizing past personal data disclosures	Service computing	✗	✗
	Lin Yuan et al. [23]	Analysis of the minimal privacy disclosure	Service computing	✓	✗
Risk	Yu et al. [24]	Modeling and measuring privacy risks	Service computing	✓	✗
	Hong et al. [25]	Privacy risk models	Ubiquitous computing	✗	✓
Obfuscation	Weiwei and Zhihong [26]	A mixed mode data obfuscation method AENDO	N/A	N/A	✗
	Bakken et al. [27]	Anonymity and desensitization of usable data sets	N/A	N/A	✗
Data oriented privacy protection	Bao et al. [28]	An efficient and practical scheme for privacy protection in the e-commerce of digital goods	N/A	N/A	✗
	Gilburd et al. [29]	k-TTP: a new privacy model for large-scale distributed environments	N/A	N/A	✗
Anonymity	Ye et al. [30]	Anonymizing classification data using rough set theory	N/A	N/A	✗
	Sweeney [31]	Achieving k-anonymity privacy protection using generalization and suppression	N/A	N/A	✗

N/A: not applicable; ✓: support; ✗: not support.

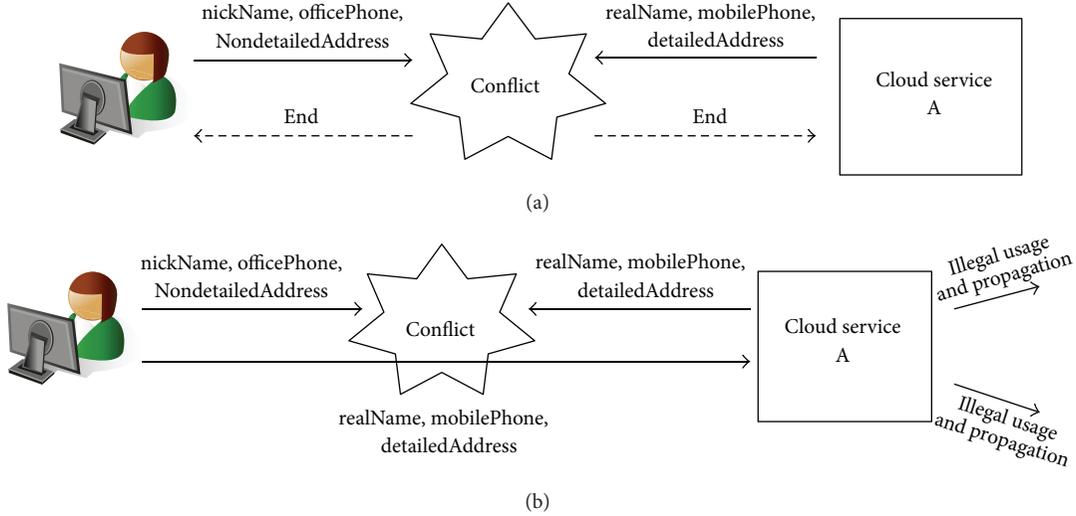


FIGURE 1: (a) Service is terminated because of privacy conflict. (b) Service proceeded with privacy information disclosed.

is decidable subset of first-order logic and formalism for representing knowledge. Description logic is also called term logic, terminology knowledge representation language, concept language, and term representation language. Description logic is composed of concepts, roles, and individuals. Complex concepts and roles can be described by simple concepts and roles.

In this paper, we build a model of the privacy negotiation between service provider and user by taking advantage of description logic, transforming the privacy conflict issue to be decidable issue of Tableau algorithm. Supposing  $A$  and  $B$  are atomic concept,  $C$  and  $D$  are concept description,  $\varphi$  and  $\varphi'$  are atomic formula,  $p$  and  $q$  represent individuals, and  $R$  and  $S$  represent atomic roles. Basic constructors include atomic negation  $\neg$ , atomic intersection  $\sqcap$ , value restriction  $\forall$ , and limited existential quantification  $\exists$ . This basic description logic is called ALC. All concept descriptions in ALC can be achieved through the following syntax rule:

$$C, D \longrightarrow A \mid \{p\} \mid \neg A \mid C \sqcap D \mid C \sqcup D \mid \forall R \cdot C \mid \exists R \cdot C \mid \perp \mid \top. \quad (1)$$

All formulas in ALC can be obtained through the following atomic formula:

$$\begin{aligned} \varphi, \varphi' \longrightarrow & C(p) \mid R(p, q) \mid \neg\varphi \mid \varphi \vee \varphi' \mid \varphi \\ & \wedge \varphi' \mid \varphi \longrightarrow \varphi \mid \text{true} \mid \text{false}. \end{aligned} \quad (2)$$

Syntax and semantics in (1) of ALC are shown in Table 2.

Tableau algorithm is an algorithm of detecting satisfiability among concepts in description logic. Since reasoning issue in description logic can be specified as satisfiability issue among concepts, most reasoners use Tableau algorithm, such as Pellet and Fact. Supposing that negative normal form of concept  $A$  is  $\text{nnf}(A)$ , notation  $[\text{path}]$  of each concept represents path of concept generated. The reasoning rule of Tableau algorithm is as follows.

- (1) Extension rule: supposing  $A$  is atomic concept and  $A \sqsubseteq B$ ,  $A^{[\text{path}]} \in \mathcal{A}(x)$ ,  $\text{nnf}(B) \notin \mathcal{A}(x)$ , then  $\mathcal{A}(x) = \mathcal{A}(x) \cup \{\text{nnf}(B)^{[\text{path}:A]}\}$ .
- (2)  $\sqcup$  rule: supposing  $C = \{C_1, C_2\}$ , if  $C_1 \sqcup C_2 \in \mathcal{A}(x)$ ,  $\{C_1, C_2\} \cap \mathcal{A}(x) = \emptyset$ , then  $\mathcal{A}(x) \rightarrow \mathcal{A}(x) \cup C$ .
- (3)  $\sqcap$  rule: supposing  $C = \{C_1, C_2\}$ , if  $C_1 \sqcap C_2 \in \mathcal{A}(x)$ ,  $\{C_1, C_2\} \notin \mathcal{A}(x)$ , then  $\mathcal{A}(x) \rightarrow \mathcal{A}(x) \cup \{C\}$ .
- (4)  $\exists$  rule: supposing  $C = \{C_1, C_2\}$ , if  $\exists S \cdot C \in \mathcal{A}(x)$ , if  $x$  does not have successor  $y$  of  $S$  that makes  $C \in \mathcal{A}(y)$ , then we add a note  $y$ , value  $\mathcal{A}(x, y) = S$ , and  $\mathcal{A}(y) = \{C\}$ .
- (5)  $\forall$  rule: supposing  $C = \{C_1, C_2\}$ , if  $\forall S \cdot C \in \mathcal{A}(x)$ , if  $x$  does not have successor  $y$  of  $S$ , and  $C \notin \mathcal{A}(x)$ , then  $\mathcal{A}(x) \rightarrow \mathcal{A}(x) \cup \{C\}$ .

#### 4.2. Privacy-Oriented Cloud Service Description Model.

Compared to traditional web services, context semantic information is considered for services in cloud computing, which improves self-adaptive and self-management ability of service and increases intelligent level. Supposing the atomic service in cloud computing is described with OWL-S, outsourcing service in cloud computing description model is defined as below.

*Definition 1* (outsourcing service metamodel). Outsourcing service metamodel can be expressed with 4 tubes, namely, Outsourcing Service Description = {Ontology, Profile, Privacy, Capability}, in which ontology is basic terminology of service description, profile describes basic information about service, such as service name, service provider, service version, and QOS, privacy mainly describes privacy related information, such as input and precondition of service, capability describes the function of service including output and result. Privacy-oriented outsourcing service model is showed in Figure 2. In this paper, we mainly focus on the privacy related information. The

TABLE 2: Syntax and semantic of ALC.

Constructor	Syntax	Semantic	Instances
Atomic concept	$A$	$A^I \subseteq \Delta^I$	Name
Atomic relationship	$R$	$R^I \subseteq \Delta^I \times \Delta^I$	hasIDNumber
Atomic negation	$\neg A$	$\Delta^I \setminus C$	$\neg$ Name
Intersection	$A \sqcap B$	$C^I \cap D^I$	Name $\sqcap$ IDNumber
Value restriction	$\forall R \cdot C$	$\{x   \exists y. \langle x, y \rangle \in R^I \wedge y \in C^I\}$	$\forall$ hasIDNumber. Name
Limited existential	$\exists R \cdot C$	$\{x   \forall y. \langle x, y \rangle \in R^I \rightarrow y \in C^I\}$	$\exists$ hasIDNumber. Name

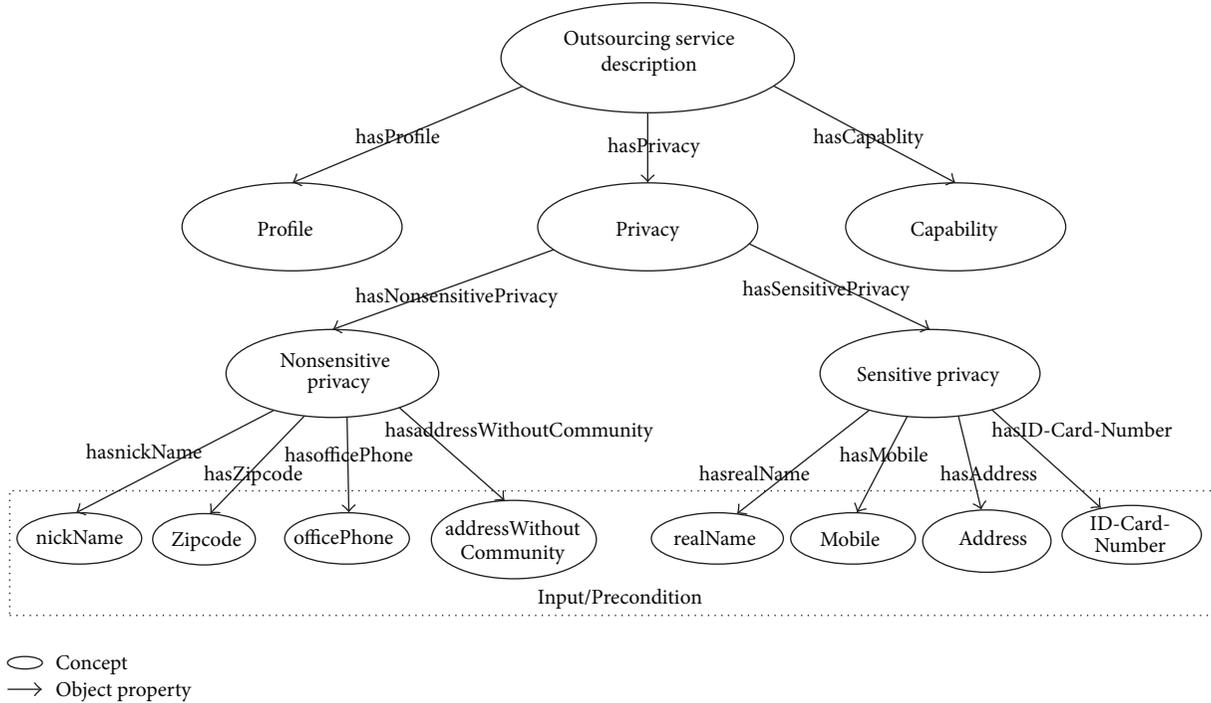


FIGURE 2: Privacy-oriented cloud service description model.

other information is omitted; for example, detailed information of profile and capability is not shown in Figure 2. We can express outsourcing service metamodel as follows:

Outsourcing Service Metamodel =  $(\exists \text{has Profile.Profile}) \sqcap (\exists \text{has Privacy.Privacy}) \sqcap (\exists \text{has Capability.Capability})$ .

*Definition 2* (privacy in the outsourcing service). Privacy can be expressed as 2 tubes, namely, Privacy  $\sqsubseteq \{\text{Input, Precondition}\}$ , in which mapping is TBox. We can express it as follows:

Privacy  $\sqsubseteq \exists \text{has input (service-metamodel, input)} \sqcap \exists \text{has precondition (service-metamodel, precondition)}$ .

#### 4.3. Service Trust Degree Metric

*Definition 3* (trust degree ( $D$ )). Trust degree is level of which service or service provider can be trusted. We can express it as  $D = \Delta(S, C, Re)$ , in which  $S$  represents security, certificating the truth and integrity of data and trustworthy of QOS,  $C$  represents capability of service or service provider to meet user security requirement, and  $Re$  represents reputation of

service or service provider regarded by user [26]. In the meantime,  $S$ ,  $C$ , and  $Re$  are attributions of trust degree.

(1) Security evaluation mainly evaluates if service has encryption, digital signature, or WSLA security, defined as follows:

$$D_1 = \begin{cases} \frac{\text{en}(s) + \text{ds}(s) + \text{ws}(s)}{3} & \text{en}(s) \vee \text{ds}(s) \\ & \forall \text{ws}(s) \in \{0, 1\} \\ 0 & \text{others.} \end{cases} \quad (3)$$

In which  $\text{en}(s)$  represents that service has encryption,  $\text{ds}(s)$  represents that service has function of digital signature, and  $\text{ws}(s)$  represents that service has WSLA security.

(2) Capability evaluation is defined through the frequency of user accessing service:

$$D_2 = \frac{|\{\text{user}_1, \text{user}_2, \dots, \text{user}_i, \dots, \text{user}_n\}|}{\sum_{u \in \text{user}_i} c(s_u)}. \quad (4)$$

In which  $\text{user}$  represents those users who access service  $s$  during the period of  $\delta$ ,  $c(s_u)$  represents counts that service be accessed in period of  $\delta$  by user  $u \in \text{user}_i$ .

(3) Reputation evaluation is evaluated by feedback from user and defined as follows:

$$D_3 = \frac{\sum_{u \in UC(s)} f(u, s, t) \times e^{-\varepsilon(c_{\text{time}} - t)}}{|UC(s)|}. \quad (5)$$

In which  $UC(s)$  represents user collection which evaluates service  $s$  in period of  $\delta$ ,  $f(u, s, t)$  represents all evaluation information from user at time  $t$  to  $s$ ,  $e^{-\varepsilon(c_{\text{time}} - t)}$  is time attenuation function while  $\varepsilon$  is attenuation factor, and  $c_{\text{time}}$  is current time.

From formulas (3), (4), and (5), we can obtain the formula calculating service trust degree:

$$D = \sum_{i=1}^3 w_i \times D_i \quad (w_i > 0). \quad (6)$$

In which  $w_i$  is weight of different trust degree attribution in service and  $D_i$  is value of different trust degree attribution. Set  $\alpha$  as threshold of user expected service trust degree. If  $D \geq \alpha$ , user will accept all privacy attribution of service, no need for further privacy conflict detection by system, or else system has to detect privacy conflict to satisfy user privacy preference.

## 5. Privacy Conflict Detection

*Definition 4* (sensitive degree). Sensitive privacy items are the items that set the level for privacy information according to user habit, scenario, and outsourcing service trust degree. Sensitive degree is a value of sensitive items. Therefore, user privacy information is classified as sensitive privacy information and nonsensitive privacy information on the basis of sensitive degree.

*Definition 5* (user privacy preference). Constraint is expressed by user based on user privacy information sensitive degree and constraint is assertion that should be satisfied by outsourcing service. Assertion is represented by  $\varphi$ . User privacy preference is assertion collection and mapped into ABox. Namely,

$$\text{privacyPreference} = \{\varphi_1, \varphi_2, \varphi_3, \dots, \varphi_n\}. \quad (7)$$

*Example 6.* When customer Tom sends request to outsourcing service  $A$  but trust degree of outsourcing service  $A$  or the provider of service  $A$  is equal or greater than threshold, namely,  $D \geq \alpha$ , under this condition, Tom discloses his real name and mobile phone as the service input or precondition. Constraint can be obtained by using privacy preference editor and can be expressed as follows:

$$\varphi_1 = \text{holdrealName}(A, \text{Tom}) \sqcap \text{holdmobilePhone}(A, 123456).$$

*Example 7.* When customer Tom sends request to outsourcing service  $A$  but trust degree of outsourcing service  $A$  or the provider of service  $A$  is less than threshold, namely,  $D < \alpha$ , under this condition, Tom will use nickname and office phone as service input or precondition, not willing to disclose

community information and mobile phone. Constraint can be expressed as follows:

$$\varphi_2 = \exists \text{holdrealname}(A, \text{Tom}) \sqcap \text{holdaddressWithoutCommunity}(A, (\text{YUDAO STREET, NANJING CITY, JIANGSU PROVINCE, CHINA})) \sqcap \text{holdOfficePhone}(A, +86-02586868666).$$

*Definition 8* (privacy items). Outsourcing service requests user to disclose minimum privacy data collection. Namely,  $\text{privacyItems} = \{pr_1, pr_2, pr_3, \dots, pr_k\}$ . From perspective of set theory, privacy items are subset of outsourcing service input and precondition. Namely,  $pr_i \subseteq (P_i, I_i), 0 \leq i \leq k$ . In which  $\text{privacyItems}$  is privacy data collection,  $pr$  is privacy data requested to be disclosed, and  $I$  and  $P$ , respectively, represent input and precondition of outsourcing services.

*Definition 9* (matching between user privacy preference and privacy item). There are two kinds of results for the matching. Detailed results are shown as follows.

(1) All services in outsourcing service collection satisfy user privacy preference.

As corresponding privacy items collection of outsourcing services to be composed,  $\text{privacyItems} = \{pr_1, pr_2, pr_3, \dots, pr_k\}$  is a programming that satisfies ABox  $\varphi$  for  $S_i$ , namely, satisfying the following formula:

$$\{\text{privacyItems} \wedge \text{service}(S_i) \wedge \langle \text{privacyItems} \rangle \varphi_i\} \wedge \Phi. \quad (8)$$

In which  $\text{service}(S_i)$  represents one outsourcing service in service collection to be composed,  $\langle \text{privacyItems} \rangle \varphi_i$  represents the matching relationship between privacy item and privacy preference constraint, and  $\Phi$  represents that all services satisfy user privacy preference; corresponding formula is  $\text{vice}(S_1) \mapsto \langle pr_1 \rangle \varphi_1 \wedge \dots \wedge \text{service}(S_k) \mapsto \langle pr_k \rangle \varphi_k$ .

(2) Some but not all services in outsourcing service collection satisfy user privacy preference.

As corresponding privacy item collection of outsourcing services to be composed,  $\text{privacyItems} = \{pr_1, pr_2, pr_3, \dots, pr_k\}$  is a programming that partly satisfies ABox  $\varphi$  for  $S_i$ , namely, satisfying the following formula:

$$\{\text{privacyItems} \wedge \text{service}(S_i) \wedge \langle \text{privacyItems} \rangle \varphi_i\} \wedge \Gamma. \quad (9)$$

In which  $\Gamma$  represents that some but not all services satisfy user privacy preference; corresponding formula is  $\text{service}(S_1) \mapsto \langle pr_1 \rangle \varphi_1 \vee \dots \vee \text{service}(S_k) \mapsto \langle pr_k \rangle \varphi_k$ .

*5.1. Privacy Conflict Detection Algorithm.* Suppose atomic service collection of service provider is  $S = \{S_1, S_2, \dots, S_n\}$ , its corresponding privacy item collection is  $\text{privacyItems} = \{pr_1, pr_2, pr_3, \dots, pr_k\}$ , and  $\varphi$  is user privacy preference assertion. The process of privacy conflict detection is shown as follows.

In the process of service composition, firstly service input and precondition are obtained from service description document OWL-S, from which privacy items of service can also be obtained. Then keep detecting privacy conflict according

to user privacy preference assertion  $\varphi$ , or extension of  $\varphi$ , until one service collection that satisfies user privacy preference assertion  $\varphi$  is found. If there is no service collection to satisfy  $\varphi$ , then service composition is stopped.

The first and the second line of Algorithm 1 are input and output, respectively. From the third to fifth line, respectively, initiate queue of service sequence to be privacy detected, queue of privacy item collection, and queue of service sequence that meet user privacy preference after detection. From the sixth line to the tenth line, enter service sequence to be privacy detected into queue and obtain trust degree value and privacy item collection of each atomic service successively. From eleventh line to twenty-first line, bind privacy item collection and trust degree value; then enter it into queue of privacy item collection and get head of queue successively detecting privacy conflict with Tableau algorithm; if there is no conflict, enter service into service queue that satisfies user privacy preference strategy, or else, rebind new service.

**5.2. Privacy Conflict Detection Framework.** There are two layers for privacy conflict detection framework.

*Privacy Conflict Predetection Layer.* The part with slash background in Figure 3 represents privacy conflict predetection layer. This part mainly implements three functions as follows.

- (1) User privacy requirement is translated into privacy preference assertion  $\{\varphi_1, \varphi_2, \varphi_3, \dots, \varphi_n\}$  by user privacy preference editor.
- (2) User comment information and Qos in service description document are evaluated by trust degree calculator, so as to obtain the trust degree value for services.
- (3) The input and precondition in service description document are captured by Xpath, and input and precondition are refined into privacy items.

At last, the privacy preference assertion, trust degree, and privacy items are saved into privacy conflict detection knowledge base.

*Privacy Conflict Detection Layer.* The part with grid background in Figure 3 represents privacy conflict detection layer.

Privacy conflict detection layer contains knowledge base and privacy conflict reasoner, in which knowledge base is made up of privacy preference assertion, trust degree, and privacy items. In this layer, privacy conflict detection for knowledge base is implemented by privacy conflict reasoner and the detection result is returned to user.

Therefore, framework of privacy conflict detection is showed in Figure 3.

## 6. Case Study and Experiment Analysis

**6.1. Case Study.** We prove the feasibility and effectiveness of our method by taking online purchase as an example. Firstly we assume the following points.

- (i) The less service required privacy items, the less probability of user privacy information to be disclosed.

- (ii) The less atomic service in service composition, the less scope of user privacy information to be propagated and the less risk of disclosure.

Therefore, in this example we make the payment terms to be cash on delivery to decrease the possibility of propagation or disclosure of user sensitive privacy information among atomic service provider, like Credit-Card-no., ID-Card-no., or Realname.

The online purchase service includes customer (Tom), cloud service composer (CSC), and three associate participants which are online purchase platform E-commerce service, seller (S), and shipper. In Figure 4, we specially depict the foundation service of E-commerce service, like cryptographic service, operation system service, and infrastructure service. Name, address, postcode, and phone are customer personal privacy data. The purchase process is as follows.

When customer sends order request to seller through CSC and E-commerce Service, E-commerce service, seller, and shipper will send privacy data request to customer through CSC and the obtained privacy data will be worked as input and precondition. Once the privacy data is obtained, seller will send goods to customer through shipper. Shipper will collect the payment and return to seller. Considering that cloud computing has distributive character and all entities in cloud computing are service, we suppose that all privacy data are encrypted with cryptographic service before being transmitted to OS service and infrastructure service. Therefore, we just focus on the use and disclosure of privacy data in outsourcing service except OS service and infrastructure service. In this paper, we design a privacy conflict detection service between customer and CSC. This service will detect the conflict between the requested privacy data of each outsourcing service and customer privacy requirement and then send feedback to CSC and customer. Detailed process is showed in Figure 4 case of online shopping.

Based on Figure 4 we form TBox of privacy items:

Ting  $\equiv$  Customer  $\sqcap$  Privacy  $\sqcap$  privacyOwner

Customer  $\equiv$  Woman  $\sqcap$  Man

Privacy  $\equiv$  ID-Card-Number  $\sqcap$  Address  $\sqcap$  Name  $\sqcap$  Phone

privacyOwner  $\equiv$  E-commerceService  $\sqcap$  Seller  $\sqcap$  Bank  $\sqcap$  cloudServiceComposer  $\sqcap$  Shipper

Name  $\equiv$  Realname  $\sqcap$  NickName

realName  $\equiv$  firstName  $\sqcap$  secondName  $\sqcap$  lastName

nickName  $\equiv$  Mr.Firstname  $\sqcup$  Ms.Firstname

Address  $\equiv$  Community  $\sqcap$  Street  $\sqcap$  City  $\sqcap$  Province  $\sqcap$  Country

AddressWithoutCommunity  $\equiv$  Address  $\sqcap$   $\forall$ hasAddress.  $\neg$  Community

Phone  $\equiv$  Mobile  $\sqcap$  officephone.

In this case, service composition participants include E-commerce service, seller, and shipper. Since E-commerce service, seller, and shipper own the same user privacy data in the business process, we will just discuss the privacy

```

(1) Input: The description document of Service OWL-S
(2) Output: The service composition satisfied the privacy preferences policy
(3) Init Queue ( $S_i$ );
(4) Init Queue ( $pr_i$ );
(5) Init Queue ( $service_i$ );
(6) EnQueue (Queue ( $S_i$ ),  $\{S_1, S_2, \dots, S_n\}$ );
(7) while (Queue ( $S_i$ )  $\neq \phi$ ) do
(8)   GetHead (Queue ( $S_i$ ),  $S_i$ )
(9)    $D = \sum_{i=1}^3 w_i \times D_i$ ; //calculating the Trust Degree of service
(10)   $pr_i = \text{XPath}(S_i, pr_i)$  //obtaining the privacy items of service
(11)  EnQueue (Queue ( $pr_i$ ),  $\langle D, pr_i \rangle$ );
(12)  while (Queue ( $pr_i$ )  $\neq \phi$ ) do
(13)    GetHead (Queue ( $pr_i$ ),  $pr_i$ );
(14)     $Taleau(pr_i, \varphi)$ ; //detecting privacy conflict between privacy preference and privacy items
(15)    if ( $service(S_i) \mapsto \langle pr_i \rangle \varphi_i = \text{true}$ ) do
(16)      EnQueue (Queue ( $service_i$ ),  $service_i$ );
(17)    else
(18)      rebindingservice; //choosing a new service
(19)    end if
(20)  end while
(21) end while

```

ALGORITHM 1

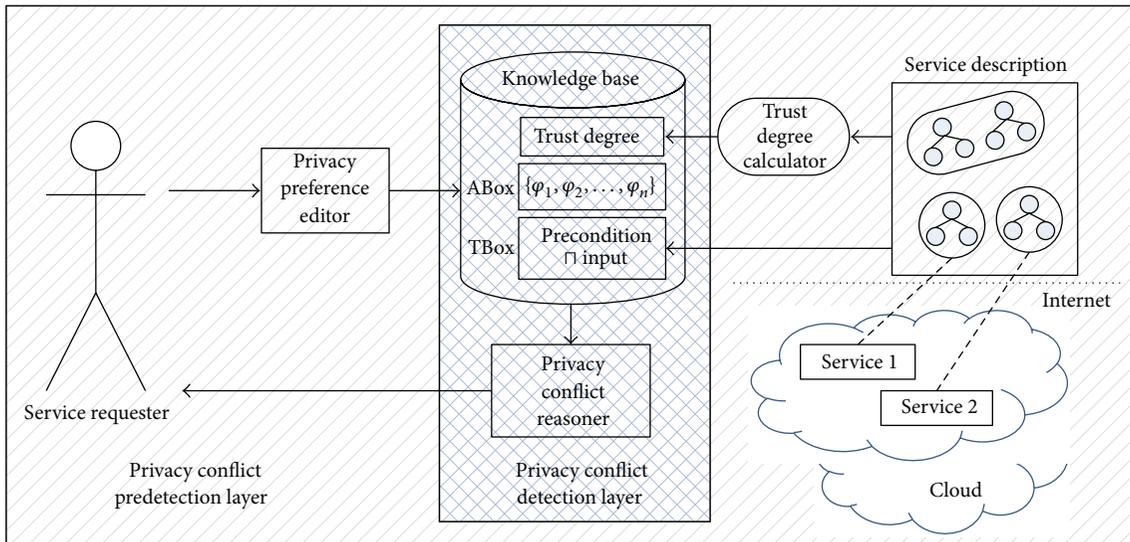


FIGURE 3: Framework of privacy conflict detection.

conflict detection for E-commerce. Detailed privacy conflict detection step is shown as follows.

*First Step.* Obtain privacy items of E-commerce service from OWL-S (outsourcing service description) and assign value to it.

realName (Changbo Ke);  
 nickName (Mr.Ke);  
 Street (YUDAO STREET);  
 City (NANJING);

Province (JIANGSU);  
 Country (CHINA);  
 Community (MINGUGONG)  
 officePhone (+86-0258686866)  $\cup$  Mobile (+86-123456789);  
 ZipCode (210016).

*Second Step.* Obtain the user privacy preference assertion  $\varphi$  from privacy conflict detection knowledge base. Namely, obtain the assertion in Abox.

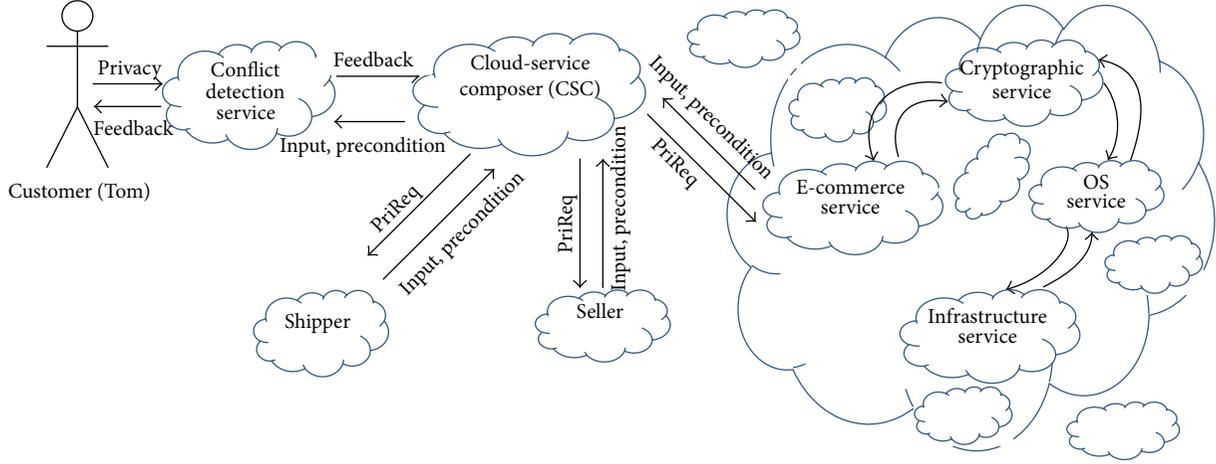


FIGURE 4: Case of online shopping.

$\varphi = \exists \text{holdRealName} (A, \text{realName}) \sqcap \forall \text{holdAddress} (A, \text{addressWithoutCommunity}) \sqcap \text{holdofficePhone} (A, \text{officePhone})$ .

*Third Step.* Detect privacy conflict, by taking advantage of privacy conflict reasoner.

- (1) Extend the nonatomic concept `AddressWithoutCommunity` with extension rule: suppose  $A$  is atomic concept and  $A \sqsubseteq B$ ,  $A^{\text{[path]}} \in \mathcal{A}(x)$ ,  $\text{nnf}(B) \notin \mathcal{A}(x)$ ; then  $\mathcal{A}(x) = \mathcal{A}(x) \cup \{\text{nnf}(B)^{\text{[path]:A}}\}$ .

We can obtain that  $\varphi = \exists \text{holdRealname.Name} (\text{Brobo}) \sqcap \text{Address} \sqcap \forall \text{holdAddress} . \neg \text{Community} (\text{YUDAO STREET, NANJING CITY, JIANGSU PROVINCE, CHINA}) \sqcap \text{holdOfficePhone} (\text{Brobo, +86-0258686866})$ .

- (2) Extend nonatomic concept `Address` with extension rule again, and we can obtain that

$\varphi = \exists \text{holdRealname.Name} (\text{Brobo}) \sqcap \text{Community} \sqcap \text{Street} \sqcap \text{City} \sqcap \text{Province} \sqcap \text{Country} \sqcap \forall \text{holdAddress} . \neg \text{Community} (\text{YUDAO STREET, NANJING CITY, JIANGSU PROVINCE, CHINA}) \sqcap \text{holdOfficePhone} (\text{Brobo, +86-0258686866})$ .

- (3) Take advantage of  $\exists$  rule of Tableau algorithm; suppose  $C = \{C_1, C_2\}$ ; if  $\exists S \cdot C \in \mathcal{A}(x)$  and  $x$  does not have successor  $y$  of  $S$  that makes  $C \in \mathcal{A}(y)$ , then add a node  $y$  and assign value  $\mathcal{A}(x, y) = S$  and  $\mathcal{A}(y) = \{C\}$ . Simplify the above formula and we can obtain that

$\varphi = \text{Name} (\text{Ke Changbo}) \sqcap \text{holdRealname} (\text{Brobo, Ke Changbo}) \sqcap \text{Community} \sqcap \text{Street} \sqcap \text{City} \sqcap \text{Province} \sqcap \text{Country} \sqcap \forall \text{holdAddress} . \neg \text{Community} (\text{YUDAO STREET, NANJING CITY, JIANGSU PROVINCE, CHINA}) \sqcap \text{holdOfficePhone} (\text{Brobo, +86-0258686866})$ .

- (4) Take advantage of  $\forall$  rule of Tableau algorithm; suppose  $C = \{C_1, C_2\}$ ; if  $\forall S \cdot C \in \mathcal{A}(x)$ , while  $C \notin \mathcal{A}(x)$ , then  $\mathcal{A}(x) \rightarrow \mathcal{A}(x) \cup \{C\}$ . Simplify the above formula and we can obtain that

$\varphi = \text{Name} (\text{Ke Changbo}) \sqcap \text{holdRealname} (\text{Brobo, Ke Changbo}) \sqcap \text{Street} \sqcap \text{City} \sqcap \text{Province} \sqcap \text{Country} \sqcap \text{holdOfficePhone} (\text{Brobo, +86-0258686866})$ .

- (5) Take advantage of  $\sqcap$  rule of Tableau algorithm; suppose (a)  $C_1 \sqcap C_2 \in \varphi(x)$  and  $x$  is not blocked directly, (b)  $\{C_1, C_2\} \notin \varphi(x)$ ; then  $\varphi(x) \rightarrow \varphi(x) \cup \{C_1, C_2\}$

$\varphi = \text{Name} (\text{Ke Changbo}), \text{holdRealname} (\text{Brobo, Ke Changbo}), \text{Street}, \text{City}, \text{Province}, \text{Country}, \text{holdOfficePhone} (\text{Brobo, +86-0258686866})$ .

- (6) Through simplifying the above formula we can obtain that

$\varphi = \text{Realname}, \text{Street}, \text{City}, \text{Province}, \text{Country}, \text{OfficePhone}$ , substitute it with value of privacy items,  $\varphi = \text{Ke Changbo, YUDAO, NANJING, JIANGSU, CHINA, +86-0258686866}$ , satisfying the formula  $\text{service}(S) \mapsto \langle pr \rangle \varphi$ , therefore no conflicts, satisfying user privacy preference assertion.

Therefore, the service composition sequence  $\{S_1, S_2, S_3\}$  satisfies the formula  $\{[pr_1, pr_2, pr_3, \dots, pr_i] \wedge \text{service}(S_i) \wedge \langle \rho \rangle \varphi_i\} \wedge \Phi$ , which also means satisfying user privacy preference.

**6.2. Experiment Analysis.** We build the ontology file “*privacy-conflict-detection.owl*” with Protégé, which is based on java language and developed by Stanford University. The conceptions and instants in the ontology are mapped into Tbox. Privacy preference assertions are defined with conceptions, items, and instants and are mapped into Abox. Tbox and Abox compose the privacy conflict detection knowledge base. We save the ontology file “*privacy-conflict-detection.owl*” to e disk test directory in local computer, then reason the ontology file with reasoner Pellet, which is developed by Mind Swap lab in University of Maryland. Pellet version number used in this experiment is V.2.3.0.

In ontology model, there are logical axioms 175 belonging to axioms 255, individuals 25, classes 33, object properties 21, and data properties 1, as shown in Figure 5

```

E:\test\pellet-2.3.0>pellet info e:\test\privacy-conflict-detection.owl
Information about file:/e:/test/privacy-conflict-detection.owl <<http://www.sema
nticweb.org/ontologies/2013/0/Ontology1358410681710.owl>>
OWL Profile = OWL 2 DL
DL Expressivity = ALR(D)
Axioms = 255
Logical Axioms = 175
GCI Axioms = 0
Individuals = 25
Classes = 33
Object Properties = 21
Data Properties = 1
Annotation Properties = 0

E:\test\pellet-2.3.0>pellet consistency e:\test\privacy-conflict-detection.owl
Consistent: Yes

E:\test\pellet-2.3.0>pellet unsat e:\test\privacy-conflict-detection.owl
Finding unsatisfiable 32 elements
Finding unsatisfiable: 100% complete in 00:00
Finding unsatisfiable finished in 00:00

Found no unsatisfiable concepts.

E:\test\pellet-2.3.0>

```

FIGURE 5: Checking results in privacy conflict detection ontology.

Firstly, we use command to detect the consistency of concept in ontology file. Command is `pellet consistency e:\test\privacy-conflict-detection.owl`. Running result is showed as red box in Figure 5, namely, consistent. It means that privacy items of service providers satisfy semantic consistency.

Secondly, we use command to detect the satisfiability between ontology concept and logic axiom in ontology file, namely, whether the relationship among ontology concepts satisfies logic axiom. Running result shown as green box in Figure 5, namely, found no unclassifiable concepts. This result means that privacy concept, owned by `privacyHolder` in ontology file, meets user privacy preference assertion  $\varphi$ . `PrivacyHolder` in ontology file is also service provider in privacy conflict detection knowledge base. Therefore, result shows that there is no conflict between user privacy preference and service provider privacy policy.

## 7. Conclusions and Future Work

In this paper, we firstly obtain input and precondition of service from service description document OWL-S in cloud computing, model service privacy item, and user privacy preference by taking advantage of knowledge base, verify the decidability of knowledge base with Tableau algorithm, and detect the conflict between service privacy item, and user privacy preference, so as to enable user to choose service collection that meets user privacy preference. We also provide privacy conflict detection algorithm. Through case study we prove the feasibility and effectiveness of our method. Further work is to negotiate between user and service provider privacy item, so as to meet both user and service provider privacy requirement.

## Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

## Acknowledgments

This work was supported by the National Natural Science Foundation of China 61272083 and 61262002, Funding for Outstanding Doctoral Dissertation in NUAU (Grant BCXJ12-14), and the Fundamental Research Funds for the Central Universities.

## References

- [1] P. Mell and T. Grance, Draft NIST working definition of cloud computing, 2009, <http://csrc.nist.gov/groups/SNS/cloud-computing/index.html>.
- [2] M. Armbrust, A. Fox, R. Griffith et al., "Above the clouds: a berkeley view of cloud computing," Tech. Rep. UCB-EECS-28, University of California, Berkeley, Calif, USA, 2009.
- [3] L. D. Brandeis and S. D. Warren, *The Right to Privacy*, edited by Steven Alan Childress, 2010.
- [4] I. Goldberg, D. Wagner, and E. Brewer, "Privacy-enhancing technologies for the internet," in *Proceedings of the 42nd IEEE International Computer Conference (COMPCON '97)*, pp. 103–109, 1997.
- [5] F. Xiao, Z. Huang, Z. Cao, J. Hu, and L. Liu, "Modeling cost-aware web services composition using PTCCS," in *Proceedings of the IEEE International Conference on Web Services (ICWS '09)*, pp. 461–468, 2009.
- [6] G. Yee and L. Korba, "Privacy policy compliance for web services," in *Proceedings of the IEEE International Conference on Web Services (ICWS '04)*, pp. 158–165, 2004.

- [7] J. Zhang, C. K. Chang, L.-J. Zhang, and P. C. K. Hung, "Toward a service-oriented development through a case study," *IEEE Transactions on Systems, Man, and Cybernetics Part A: Systems and Humans*, vol. 37, no. 6, pp. 955–969, 2007.
- [8] S. Pearson, "Taking account of privacy when designing cloud computing services," in *Proceedings of the ICSE Workshop on Software Engineering (CLOUD '09)*, IEEE Computer Society, Vancouver, Canada, 2009, HP Labs Technical Report, HPL-54.
- [9] S. Pearson and A. Charlesworth, "Accountability as a way forward for privacy protection in the cloud," HP Labs Technical Report HPL-178, 2009.
- [10] S. Pearson, V. Tountopoulos, D. Catteddu et al., "Accountability for cloud and other future Internet services," in *Proceedings of the IEEE 4th International Conference on Cloud Computing Technology and Science (CloudCom '12)*, pp. 629–632, 2012.
- [11] I. Roy, H. E. Ramadan, S. T. V. Setty et al., "Airavat: security and privacy for MapReduce," in *Proceedings of the 7th Usenix Symposium on Networked Systems Design and Implementation*, M. Castro, Ed., pp. 297–2312, USENIX Association, San Jose, Calif, USA, 2010.
- [12] K. D. Bowers, A. Juels, and A. Oprea, "Proofs of retrievability: theory and implementation," in *Proceedings of the ACM Workshop on Cloud Computing Security with the 16th ACM Computer and Communications Security Conference (CCS '09)*, R. Sion, Ed., pp. 43–54, ACM, New York, NY, USA, 2009.
- [13] R. Hamadi, H. Y. Paik, and B. Benatallah, "Conceptual modeling of privacy-aware web service protocols," in *Proceedings of the 19th International Conference on Advanced Information System Engineering (CAiSE '07)*, pp. 233–248, 2007.
- [14] N. Guermouche, S. Benbernou, E. Coquery, and M.-S. Hacid, "Privacy-aware web service protocol replaceability," in *Proceedings of the IEEE International Conference on Web Services (ICWS '07)*, pp. 1048–1055, 2007.
- [15] K. Mokhtari, S. Benbernou, M. Hacid, E. Coquery, F. Leymann, and M. Said, "Verification of privacy timed properties in web service protocols," in *Proceedings of the IEEE International Conference on Services Computing (SCC '08)*, pp. 593–594, 2008.
- [16] L. LinYuan, Z. Haibin, H. Zhiqiu, and X. Dongqing, "Minimal privacy authorization in web services collaboration," *Computer Standards & Interfaces*, vol. 33, no. 3, pp. 332–343, 2011.
- [17] A. Barth, J. C. Mitchell, A. Datta, and S. Sundaram, "Privacy and utility in business processes," in *Proceedings of the 20th IEEE Computer Security Foundations Symposium (CSFS '07)*, pp. 279–294, 2007.
- [18] Z. Wei, M. Kang, D.-N. Jia, B. Yin, and W. Zhou, "Research on privacy-protection policy for pervasive computing," *Chinese Journal of Computers*, vol. 33, no. 1, pp. 128–138, 2010.
- [19] H. Zhu and M. C. Zhou, "Role-based collaboration and its kernel mechanisms," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 36, no. 4, pp. 578–589, 2006.
- [20] K. El-Khatib, "A privacy negotiation protocol for web services," in *Workshop on Collaboration Agents: Autonomous Agents for Collaborative Environments*, pp. 85–92, Halifax, 2003.
- [21] Y. Zhang and D.-G. Fen, "Parsimonious semantic trust negotiation," *Chinese Journal of Computers*, vol. 32, no. 10, pp. 1989–2003, 2009.
- [22] J. Kolter, M. Netter, and G. Pernul, "Visualizing past personal data disclosures," in *Proceedings of the International Conference on Availability, Reliability, and Security (ARES '10)*, pp. 131–139, 2010.
- [23] L. LinYuan, Z. Haibin, and H. Zhiqiu, "Analysis of the minimal privacy disclosure for web services collaborations with role mechanisms," *Expert Systems with Applications*, vol. 38, no. 4, pp. 4540–4549, 2011.
- [24] T. Yu, Y. Zhang, and K. J. Lin, "Modeling and measuring privacy risks in QoS web services," in *Proceedings of the 8th IEEE International Conference on E-Commerce Technology, 3th IEEE International Conference on Enterprise Computing, E-Commerce and E-Services (EEE '06)*, p. 4, 2006.
- [25] J. I. Hong, J. D. Ng, S. Lederer, and J. A. Landay, "Privacy risk models for designing privacy-sensitive ubiquitous computing systems," in *Proceedings of the 5th Conference on Designing Interactive Systems: Processes, Practices, Methods, and Techniques*, pp. 91–100, 2004.
- [26] N. Weiwei and C. Zhihong, "Clustering-oriented privacy-preserving data publishing," *Knowledge-Based Systems*, vol. 35, pp. 264–270, 2012.
- [27] D. E. Bakken, R. Parameswaran, D. M. Blough, A. A. Franz, and T. Y. J. Palmer, "Data obfuscation: anonymity and desensitization of usable data sets," *IEEE Security & Privacy*, vol. 2, no. 6, pp. 34–41, 2004.
- [28] F. Bao, R. Deng, and P. Feng, "An efficient and practical scheme for privacy protection in the e-commerce of digital goods," in *Information Security and Cryptology—ICISC 2000*, pp. 162–170, 2001.
- [29] B. Gilburd, A. Schuster, and R. Wolff, "k-TTP: a new privacy model for large-scale distributed environments," in *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 563–568, ACM, 2004.
- [30] M. Ye, X. Wu, X. Hu, and D. Hua, "Anonymizing classification data using rough set theory," *Knowledge-Based Systems*, vol. 43, pp. 82–94, 2013.
- [31] L. Sweeney, "Achieving k-anonymity privacy protection using generalization and suppression," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 10, no. 5, pp. 571–588, 2002.